Assignment 2

OOP Lab

Section V Total Marks: 20

Instructor: Md Ajwad Akil

June 27, 2023

1 Robot Simulator Design - 20

You are to create a robot simulator using the concepts of inheritance, method overriding, and static.

1.1 Problem Description:

First, you need to create the **Robot** class. Mind that this is not a public class. Follow the below instructions:

1. Create a **Robot** class with a name (String type), yearBuilt (Integer type), *static* robotCounter (Integer type) and four double variables, namely x1, x2, y1, and y2, which represent 2D coordinates.
2. Create a constructor for the class to set the values of the name and year-built variables.
3. Create appropriate getters for name and yearBuilt.
4. Create a single **setCoordinates** method for setting the four coordinates together. Remember, you are to create **one** setter for the four coordinates.
5. Write a method with the signature ***calculateDistanceTraveled(int noOfMovements)*** that returns double. Initially, it should print **"This method will be overridden!"** and return 0, as other classes will override this.
6. Write a method with signature ***display()*** that should return nothing. If the name of your robot is **Tina** and the year built is **2023**, then it should print the following lines:

   Robot: Tina
   Year Built: 2023

   This method, too, will be overridden by subclasses of robots.
7. Your **robotCounter** variable should be used to count the number of robots (more specifically, robot objects) you created. You cannot create any new method to count the number of robots that have been created. Think of how you can use the idea of a static variable to achieve this.

Create the Subclass **RacingBot** Class. Mind that this is not a public class. Follow the below instructions:

1. Create a **RacingBot** class that inherits the **Robot** class and has variables initialVelocity (double type), distanceTravelTime (double type), and write the appropriate constructor with the concept of **super()** to set these variables and the previous variables of the **Robot** class.
2. Override the ***display()*** method. Let's consider the name of your Robot is **Tina** and **initialVelocity** is 2.0, and **distanceTravelTime** is 2.0. Besides **calling the display method** of the superclass, you should also add another line to the method:

   With initial velocity of 2.0 Tina took 2.0 unit of time to move.

   So the total information printed by this method is:

Robot: Tina
Year Built: 2023
With initial velocity of 2.0 Tina took 2.0 unit of time to move.

3. Override the ***calculateDistanceTraveled(int noOfMovements)*** for this class. Here are the steps to implement this method:

   - You need to calculate the total Euclidean distance for **noOfMovements** and return the double value. For ***each*** movement, the formula for Euclidean distance is:

   $$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

   Using this, calculate the total distance traveled.
   - Use ***Math.sqrt()*** for the square root and ***Math.round()*** with the answer for rounding off the value. *(Hint: Think of how to calculate a thing repeatedly.)*

4. Write a method ***calculateAcceleration(double acceleretationTime, int noOfMovements)*** that takes these two arguments as shown in the method signature. Here are the steps to implement this method:

   - You need to calculate the final velocity by calling the method ***calculateDistanceTraveled(noOfMovements)*** and dividing the value by ***distanceTravelTime*** which is a variable of this class.
   - Use the following formula to calculate the acceleration:

   $$d = (v - u)/t \tag{2}$$

   Where

   v = Final velocity that you calculated
   u = Initial velocity which is a variable of the class
   t = The acceleration time you passed as arguments

   - Use ***Math.round()*** with the answer for rounding off the value.

Create the Subclass **ServiceRobot** Class. Mind that this is not a public class. Follow the below instructions:

1. Create a **ServiceRobot** class that inherits the **Robot** class and has variables batteryCapacity (double type), chargingRate (double type), and write the appropriate constructor with the concept of **super()** to set these variables and the previous variables of the **Robot** class.

2. Override the ***display()*** method. Let's consider the name of your Robot is **Tina** and **batteryCapacity** is 200.0, and **chargingRate** is 2.4. Besides **calling the display method** of the superclass, you should also add another line to the method. The line is:

   Battery Capacity: 200.0 Ah with a charging rate of 2.4

   So the total information printed by this method is:

   Robot: Tina
   Year Built: 2023
   Battery Capacity: 200.0 Ah with a charging rate of 2.4

3. Override the ***calculateDistanceTraveled(int noOfMovements)*** for this class. Here are the steps to implement this method:

- You need to calculate the total Manhattan distance for **noOfMovements** and return the double value. For ***each*** movement, the formula for Manhattan distance is:

$$d = | (x_1 - x_2) | + | (y_1 - y_2) | \tag{3}$$

  Using this, calculate the total distance traveled.
- Use Use ***Math.abs()*** for the absolute value and ***Math.round()*** with the answer for rounding off the value.

4. Write a method named ***calculateTimeToRecharge()*** where you return the value obtained by dividing **batteryCapacity** by **chargingRate**. Make sure to use ***Math.round()*** with the answer for rounding off the value.

1.2 Testing the program

Please follow the below steps:

1. Create a **RacingBot** object and set the values in the following way:
   - Name: FerrariBOT
   - YearBuilt: 2016
   - InitialVelocity: 2.0
   - distanceTravelTime: 2.0
2. Create a **ServiceRobot** object and set the values in the following way:
   - Name: WallE
   - YearBuilt: 2009
   - batteryCapacity: 200
   - chargingRate: 2.4
3. Take **numOfMovements** as input from the user.
4. For **RacingBot** object, print the distance traveled when **numOfMovements** is given as 5 as input by calling the appropriate method.
5. For **RacingBot** object, print the acceleration when **numOfMovements** is given as 5 as input and ***accelerationTime*** is **10** by calling the appropriate method.
6. Display the information of **RacingBot** object by calling the appropriate method.
7. For **ServiceRobot** object, print the distance traveled when **numOfMovements** is given as 5 as input by calling the appropriate method.
8. For **ServiceRobot** object, print the time to recharge by calling the appropriate method.
9. Display the information of **ServiceRobot** object by calling the appropriate method.
10. You should create a ***static method called printTotalRobots()*** that prints the total number of robots of **Robot** class. You are to print:

    Total robots created: 2
11. For the output of the program, check the *output.txt* file. The output should match if all the methods have been implemented correctly.

2 Marks Distribution

Section	Marks
Robot Class	3
RacingBot Class	7
ServiceRobot Class	7
RobotSimulation Class(Main Method)	3
Total	20

Please Follow the guidelines below:

- Please solve the problems in the *Java* language and by creating appropriate classes in the same Project.
- Zip the **WHOLE INTELLIJ PROJECT** so I can directly run it. During zipping, you should name the folders the following way: studentid_name_oop_2.zip. If your ID is 12345 and your name is akil, the zip file should be **12345_akil_oop_2.zip**.
- **DO NOT COPY from the internet, seniors, batchmates, or any other sources. You are always welcome to discuss and find the solutions together, but you must write your own code. If found out, there will be *-100%* marks reduction.**
- **DO NOT PUT** the question in **chatGPT** and ask it to write the answer. As you have seen, the result of using this method. You may use the tool to learn more about the concepts or take additional help (such as how to use certain methods or how to do certain concepts) but do not directly use it to write code for the problem. If found out, there will be *-100%* marks reduction.
- For any query, ask in class or email me at ajwad@cse.uiu.ac.bd or call me by phone: **01759099000**